Fig. 2 shows the spectra of the gain and the noise figure of the amplifier measured with a small signal probe of –30dBm. During the measurement, the amplifier was saturated with three saturation lights (whose wavelengths were 1540, 1580 and 1600nm, respectively). The total power of the signal probe and the saturation light was –14dBm at the amplifier input. The 3dB bandwidth of the total gain was 75nm (1531–1606nm) with a peak gain of 19.8dB. The dips in the gain spectrum around 1550 and 1575nm can be eliminated by optimising the equaliser profile [5]. The noise figure was from 4.9 to 7.4dB (from 3.8 to 6.3dB at the EDFF input) within the 3dB band. A further increase in the 3dB bandwidth can be expected on optimising the pump wavelength of the FP-LDs.

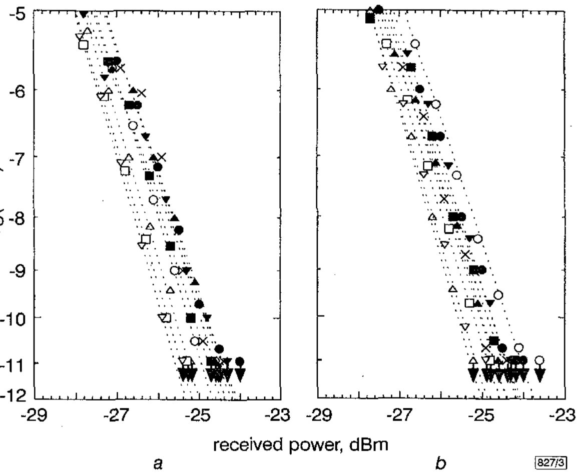

Fig. 3 *Bit error rate against received power at APD input*

a Before transmission
b After 170km transmission
○ 1531nm ● 1568nm
△ 1540nm ▲ 1578nm
□ 1550nm ■ 1588nm
▽ 1561nm ▼ 1597nm
× 1606nm

We demonstrated 9 × 2.5Gbit/s WDM transmission using our amplifier as an in-line amplifier. Shorter wavelength (1531, 1540, 1550 and 1561nm) and longer wavelength (1568, 1578, 1588, 1597 and 1606nm) NRZ signals were separately generated and multiplexed, and then launched into the transmission line comprising two spans of 85km DSF. The signal power was –4dBm per channel at the DSF input. After passing through pre-amplifiers and 1nm optical bandpass filters, the transmitted signals were detected by an APD optical receiver [6]. Resultant bit error rates (BERs) before and after transmission are plotted in Figs. 3*a* and *b*. Error-free operation was confirmed and no significant power penalties were observed for all channels.

In conclusion, the widest 3dB gain bandwidth of 75nm (1531–1606nm) and low noise figures under 7.4dB were achieved by combining a gain-flattened EDFFA with an internal Raman amplifier. Error-free operation has been confirmed in an in-line amplifier configuration. By selecting the dispersion of the DCF, ultra-wideband amplifiers will have the additional ability to compensate for fibre dispersion.

Acknowledgment: The authors would like to thank H. Yoshimura and I. Yamashita for their continuous encouragement.

© IEE 1998 *9 February 1998*
Electronics Letters Online No: 19980599

S. Kawai, H. Masuda, K.-I. Suzuki and K. Aida (*NTT Optical Network Systems Laboratories, 1-1 Hikari-no-oka, Yokosuka, Kanagawa, 239-0847 Japan*)

References

1 MASUDA, H., KAWAI, S., SUZUKI, K.-I., and AIDA, K.: 'Wideband, gain-flattened, erbium-doped fibre amplifiers with 3dB bandwidths of >50nm', *Electron. Lett.*, 1997, **33**, pp. 1070–1072

2 MORI, A., OHISHI, Y., YAMADA, M., ONO, H., NISHIDA, Y., OIKAWA, K., and SUDO, S.: '1.5μm broadband amplification by tellurite-based EDFAs'. OFC '97, 1997, Paper PD1

3 SUN, Y., SULHOFF, J.W., SRIVASTAVA, A.K., ZYSKIND, J.L., STRASSER, T.A., PEDRAZZANI, J.R., WOLF, C., ZHOU, J., JUDKINS, J.B., ESPINDOLA, R.P., and VENGSARKAR, A.M.: '80nm ultra-wideband erbium-doped silica fibre amplifier', *Electron. Lett.*, 1997, **33**, pp. 1965–1967

4 MASUDA, H., KAWAI, S., SUZUKI, K.-I., and AIDA, K.: '75nm 3dB gain-band optical amplification with erbium-doped fluoride fibre amplifiers and distributed Raman amplifiers in 9 × 2.5 Gbit/s WDM transmission experiment'. ECOC '97 PDP, 1997, pp. 73–76

5 WYSOCKI, P.F., JUDKINS, J.B., ESPINDOLA, R.P., ANDREJCO, M., and VENGSARKAR, A.M.: 'Broad-band erbium-doped fibre amplifier flattened beyond 40nm using long-period grating filter', *IEEE Photonics. Technol. Lett.*, 1997, **9**, pp. 1343–1345

6 SUZUKI, K.-I., MASUDA, H., KAWAI, S., AIDA, K., and NAKAGAWA, K.: 'Bidirectional 10-channel 2.5Gbit/s WDM transmission over 250km using 76nm (1531–1607nm) gain-band bidirectional erbium-doped fibre amplifiers', *Electron. Lett.*, 1997, **33**, pp. 1967–1968

Combining adaptive sigmoid packet and trace neural network for fast invariance-learning

Han Chuan Peng, Li Feng Sha, Qiang Gan and Yu Wei

By replacing the commonly used sigmoid neuron activation function with an adaptive sigmoid packet, the invariance extraction of a trace neural network can be effectively enhanced and speeded up for fast varying sample sequences. The performance is compared with several existing models.

Network and algorithm: One of the main problems in neural networks and machine learning is how to design effective and fast algorithms for invariance extraction [1]. Recently, a trace neural network (TNN) has been proposed to track the variation of input sample sequences [1, 2] and to self-organise to produce significant representations [2]. However, TNN suffers from instability when input samples vary extremely from one to another [1, 2]. At the same time, some existing networks, such as the back-propagation (BP) network [3] and quantum neural network (QNN) [4], cannot learn invariance at high speed with good accuracy. In this Letter, a new learning scheme using a sigmoid packet as the neuron activation function is introduced into TNN to attack the problem.

A sigmoid packet f is defined as the linear combination of a set of sigmoid functions $\{s_n, n = 1, 2, ..., N\}$ with different amplitudes $\{h_n\}$, slopes $\{a_n\}$ and shifts $\{b_n\}$:

$$f(net_j) = \sum_{n=1}^{N} h_n s_n = \sum_{n=1}^{N} \frac{h_n}{1 + \exp(-a_n \cdot net_j + b_n)} \quad (1)$$

where net_j is the weighted sum of inputs to the jth neuron. This sigmoid packet is used as the neuron activation function instead of the commonly used sigmoid function in the multilayer perceptron, and during learning, all parameters $\{h_n, a_n, b_n\}$ can be adjusted for adaptive shape-refining. Note that several slowly varying areas are introduced in the middle of the activation function to reduce the sensitivity and instability of the neuron to the varying input. This can be deduced as an adaptive version of the small-derivative rule for neural network design [5] to improve the generalisation of neural networks. These slowly varying areas can also be examined as quantum stairs [4] which merge fuzzy logic principles to detect and identity some uncertainties in pattern recognition problems.

Consider an M-class pattern recognition problem treated by a three-layer feedforward TNN, which contains a linear input layer (IL) with I neurons, a nonlinear hidden layer (HL) with J neurons for feature extraction, and a nonlinear output layer (OL) with M common sigmoid neurons for classification. HL consists of neurons employing the sigmoid packet activation function described by $y_j = f(\Sigma_{i=1}^{I} w_{ji}x_i)$, where x_i, is the ith input and w_{ji} is the connection weight from the ith input neuron to the jth neuron in HL. Define the trace of the jth neuron corresponding to the mth class of input patterns as $T_{mj}(t) = \eta T_{mj}(t - 1) + (1 - \eta)y_j(t)$ $(0 < \eta < 1)$, where η is the trace factor. Then, the learning of TNN can be formulated as the minimisation of the cost function $E = SEF + r \cdot MEF$ ($r \geq 0$) [2], where r is a factor to balance the self-influence of a neuron and the mutual-influence from other neurons. The self-energy function (SEF) is defined as $SEF(t) = 0.5\Sigma_{m=1}^{M} \Sigma_{j=1}^{J} [T_{mj}(t-1) - y_j(t)]^2$ and the mutual energy function (MEF) is defined as $MEF(y) = 0.5\Sigma_{j=1}^{J} mef[\, y_j(t)]$, where $mef[.]$ is generally an even function which is monotonically increasing on the positive semi-axis and is able to produce meaningful sparse codes of input patterns via mutual inhibition between neurons [2]. In this Letter, $mef(y_j) = 1n(1 + y_j^2)$ will be used [2, 6]. To make this TNN into a better adaptive and self-organising learning machine for unstable input pattern sequences, the following supervised self-organising (SSO) learning scheme of HL utilising the sigmoid packet is proposed in the ordinary gradient-descent way:

$$\Delta w_{ji}(t) = K_j \cdot \left[\sum_{n=1}^{N} a_n h_n s_n (1 - s_n) \right] \cdot x_i \qquad (2)$$

$$\begin{aligned} \Delta h_n(t) &= K_j \cdot s_n \\ \Delta a_n(t) &= K_j \cdot h_n s_n (1 - s_n) \cdot net_j \qquad (3) \\ \Delta b_n(t) &= K_j \cdot h_n s_n (s_n - 1) \end{aligned}$$

$$K_j = -\alpha\{[(1-\beta)(y_j - T_{mj} + r \cdot y_j/(1 + y_j^2)] + \beta\Delta_j\}$$
$$(0 < \alpha \ll 1,\ 0 \leq \beta \leq 1) \qquad (4)$$

where α is a learning factor, and β is a coefficient for balancing the effects of the unsupervised trace signal and the supervised error signal Δ_j which is back-propagated from OL to the jth neuron of HL using the BP algorithm. Apparently, SSO is a unified learning scheme. If β = 0, SSO is called a quantum trace neural net (QTNN) and will reduce to a TNN when the common sigmoid activation function is employed in HL. If β = 1, SSO becomes a pure QNN. Furthermore, when β = 1 and the common sigmoid activation function is ueed in HL, SSO is a BP net. The output layer OL is trained using the BP algorithm.

Simulation: In solving many realistic invariance learning problems, the neural net must produce good generalisation with a small number of training samples. This can be reflected as the contradiction between learning accuracy and average learning sample sequence length. The QTNN can provide a good compromise in fast invariance learning. Because the problem of unconstrained handwritten digit recognition [3] is a typical example of fast-varying input samples, it is used to verify the above algorithm.

Table 1: Performance comparison of several models

Network	Average learning sequnce length	Error	
		$r = 0$	$r = 0$ 1
		%	%
QTNN	~1 × 10⁴	3.12	0.90
TNN	~5 × 10⁴	6.82	1.03
QNN	~1 × 10⁴	3.48	
BP net	~5 × 10⁴	0.90	

A database from the CENPARMI lab in Canada, with 6000 handwritten digits (16 × 16 pixel binary images), is employed. The numbers of neurons in IL, HL and OL are set to be 20 × 20 (4 pixels for the blank border), 16 × 16, and 10, respectively. Each neuron in HL is connected to a local input area in IL using a 7 × 7 weight mask. 50 samples per class are used for training, while the whole database is used for testing. The initial parameters in QTNN and QNN are chosen as follows: $N = 2$, $a_1 = a_2 = 6.68$, $b_1 = 6$, $b_2 = -6$, $h_1 \equiv h_2 \equiv 0.5$. All the weights are uniformly initialised on [–0.2, 0.2]. Learning will stop when the learning accuracy is >95%. Table 1 gives the learning and testing performance of the QTNN, TNN, QNN, and BP net on the same database. The average learning sequence length in training is the product of training epoch, sample number per class, and the number of classes M. Because input sequences vary greatly, the error between the actual recognition rate in testing and the preset recognition rate of 80% can be a useful measure to compare the generalisation of different models. The ascendancy of the QTNN is obvious: it achieves a smaller error in testing than TNN or QNN, while its learning time is much less than TNN or BP net. The better performance of QTNN compared to TNN confirms the effectiveness of the adaptive sigmoid packet neuron activation function. It can also be noticed that better results are obtained when $r = 0.1$ than when $r = 0$. This corroborates the importance of MEF and is in accordance with the previous discussion by Peng [2].

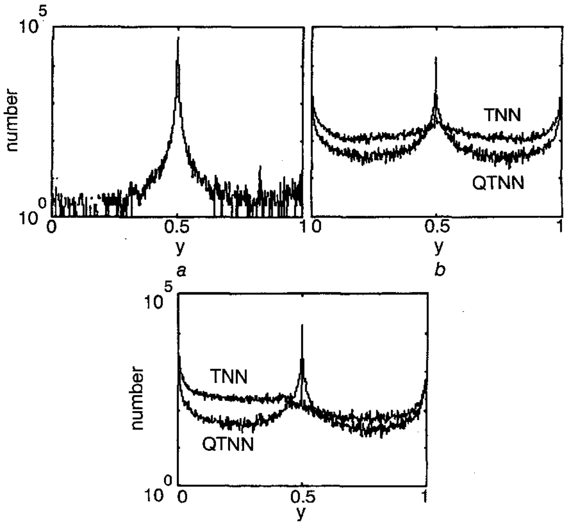

Fig. 1 *Distribution of representations in HL before and after learning*
a Initial distribution
b Final distribution with $r = 0$
c Final distribution with $r = 0.1$

The distribution of representations in HL for all training samples is shown in Fig. 1. Fig. 1*a* is the semi-logarithm histogram of outputs of HL at the beginning. There is no peak around $y = 0$ or $y = 1$. However, after learning, two additional peaks appear beside the original one in $y = 0.5$, as shown in Fig. 1*b* and *c*. It can be noticed that the adaptive sigmoid packet effectively decreases the distribution on (0, 0.5) and (0.5, 1), makes these peaks clearer, and produces sparser representations. These new peaks can naturally suggest that the sigmoid packet activation function adaptively refines its shape to form several better saturation areas to segment the representation space. Because sparser representations result in better classification performance, to some extent the above phenomenon can explain why the QTNN outperforms TNN and QNN in both recognition rate and learning speed.

Conclusion: The adaptive sigmoid packet as a neuron activation function and the corresponding learning algorithm are proposed to enhance the invariance extraction of trace neural network. The performance of the QTNN, especially the fast invariance learning capacity from a small training sample set, is better than for a TNN, a QNN, or a BP network.

© IEE 1998 *23 February 1998*
Electronics Letters Online No: 19980667

Han Chuan Peng, Li Feng Sha, Qiang Gan and Yu Wei (*Chien-Shiung Wu Laboratory, Department of Biomedical Engineering, Southeast University, Nanjing 210096, People's Republic of China*)

E-mail: phc@seu.edu.cn

Qiang Gan: Currently with Department of Electronics and Computer Science, University of Southampton, Southamptron, United Kingdom

References

1 WALLIS, G.: 'Using spatio-temporal correlations to learn invariant object recognition', *Neural Netw.*, 1996, **9**, (9), pp. 1513–1519

2 PENG, H.-C., GAN, Q., SHA, L.-F., and WEI, Y.: 'Using sparse trace neural network to learn temporal-spatial invariance'. 1998 Int. Joint Conf. on Neural Networks, Alaska, 1998

3 GAN, Q., and SUEN, C.Y.: 'Neural networks for handwritten character recognition' *in* 'Fuzzy logic and neural network handbook' (McGraw-Hill Book Company, 1996), pp. 16.1–16.6

4 PURUSHOTHAMAN, G., and KARAYIANNIS, N.: 'Quantum neural networks (QNN's): inherently fuzzy feedforward neural networks', *IEEE Trans. Neural Netw.*, 1997, **8**, (3), pp. 679–693

5 GAN, Q.: 'A neural network classifier with valid generalization performance', *J. Southeast University*, 1994, **24**, pp. 67–72 (suppl.)

6 OLSHAUSEN, B., and FIELD, D.: 'Emergence of simple-cell receptive field properties by learning a sparse code for natural images', *Nature*, 1996, **381**, pp. 607–609

Differential Hebbian-type learning algorithms for decorrelation and independent component analysis

Seungjin Choi

Differential learning algorithms for decorrelation and independent component analysis (ICA) are presented. It is shown that the proposed differential Hebbian-type learning algorithms are able to successfully decorrelate the non-zero mean-valued data without any preprocessing. Differential learning is also applied for independent component analysis (ICA) so that non-zero mean-valued source signals can be recovered without any preprocessing. It is demonstrated that modified ICA algorithms using differential learning have a superior performance compared to conventional ICA algorithms for the case where the mean values of source signals are non-zero and are changing.

Introduction: Independent component analysis (ICA) or blind source separation (BSS) is a fundamental problem encountered in many applications such as communications, sonar, image processing, and some biomedical applications. In ICA or BSS, the m dimensional vector of measured signals, $\mathbf{x}(t) = [x_1(t) \cdots x_m(t)]^T$ is assumed to be generated from an (unknown) n dimensional vector of source signals, $\mathbf{s}(t) = [s_1(t) \cdots s_n(t)]^T$ through an (unknown) linear generative model, i.e.

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) \qquad (1)$$

where $\mathbf{A}$ is an $(m \times n)$ mixing matrix, $(m \geq n)$. Source signals $\mathbf{s}(t)$ are assumed to be spatially independent and temporally IID.

In ICA or BSS, the vectors of the measured signal $\mathbf{x}(t)$ are processed by a linear feedforward neural network whose output $\mathbf{y}(t)$ is given by

$$\mathbf{y}(t) = \mathbf{W}(t)\mathbf{x}(t) \qquad (2)$$

The connection weight matrix $\mathbf{W}(t)$ is updated such that the global transformation $\mathbf{G}(t) = \mathbf{W}(t)\mathbf{A}$ converge to $\mathbf{G}(\infty) = \mathbf{P}\Lambda$ (generalised permutation matrix) as $t \to \infty$, where $\mathbf{P}$ is a permutation matrix and Λ is an nonsingular diagonal matrix.

Since Jutten and Herault's proposal [1] for an adaptive solution to ICA or BSS, a variety of approaches [2–5] have been proposed. One widely-used learning algorithm for ICA has the form of

$$\mathbf{W}(t+1) = \mathbf{W}(t) + \eta_t\{\mathbf{I} - \mathbf{f}(\mathbf{y}(t))\mathbf{y}^T(t)\}\mathbf{W}(t) \qquad (3)$$

where $\mathbf{f}(\mathbf{y}(t)) = [f_1(y_1(t)) \cdots f_n(y_n(t))]^T$ is an elementwise nonlinear function depending on the probability distribution of source signals $\mathbf{s}(t)$. $\eta_t > 0$ is a learning rate. It is known that $f_i(y_i(t)) = y_i^3(t)$ is a good choice for the sub-Gaussian source signal $s_i(t)$ and $f_i(y_i(t)) = \tanh(\beta y_i(t))$ (for $\beta > 2$) for the super-Gaussian source signal $s_i(t)$. Note that for decorrelation (second-order), the elementwise function $\mathbf{f}(\mathbf{y}(t))$ is chosen as a linear function, i.e. $\mathbf{f}(\mathbf{y}(t)) = \mathbf{y}(t)$ [6].

In most of the literature, zero-mean source signals are assumed. When the mean of the source signals is not zero, it is necessary to preprocess the data to eliminate the estimated mean value from the data. However, if the mean values of the source signals are changing, preprocessing would degrade the performance of the algorithm eqn. 3 since we do not know when the mean values of the source signals change. In this Letter, we present differential learning algorithms for decorrelation and ICA which are able to decorrelate the measured signals and separate mixtures of source signals successfully when the mean values of the source signals are changing.

Differential learning algorithms for decorrelation and ICA: The proposed differential learning algorithms for decorrelation (whitening) and ICA are inspired by the covariance learning law [7]. The modified decorrelation algorithm using differential learning law is described by

$$\mathbf{W}(t+1) = \mathbf{W}(t) + \eta_t\{\mathbf{I} - \Delta\mathbf{y}(t)\Delta\mathbf{y}^T(t)\}\mathbf{W}(t) \qquad (4)$$

where

$$\Delta\mathbf{y}(t) = \mathbf{y}(t) - \mathbf{y}(t-1) \qquad (5)$$

For successful decorrelation when source signals $\mathbf{s}(t)$ have non-zero mean, the stationary points of the averaged version of eqn. 4 should satisfy

$$E\{[\mathbf{y}(t) - \bar{\mathbf{y}}(t)][\mathbf{y}(t) - \bar{\mathbf{y}}(t)]^T\} = \mathbf{0} \qquad (6)$$

where $\bar{\mathbf{y}}(t)$ is the mean of $\mathbf{y}(t)$. It can easily be seen that without any preprocessing, the stationary points of the algorithm in eqn. 3 (with $\mathbf{f}(\mathbf{y}(t)) = \mathbf{y}(t)$) do not satisfy eqn. 6 since the algorithm in eqn. 3 is the correlation learning law.

To understand the algorithm in eqn. 4, we consider the stationary points of the averaged version of this algorithm. We introduce the global transformation matrix $\mathbf{G} = \mathbf{W}\mathbf{A}$ to relate the network output $\mathbf{y}(t)$ to source signals $\mathbf{s}(t)$. Then, we have

$$\mathbf{y}(t) = \mathbf{G}\mathbf{s}(t) \qquad (7)$$

The mean of $\mathbf{s}(t)$ is defined by $\bar{\mathbf{s}}(t)$. Then, we have

$$\mathbf{s}(t) = \tilde{\mathbf{s}}(t) + \bar{\mathbf{s}}(t) \qquad (8)$$

where $E\{\tilde{\mathbf{s}}(t)\} = 0$. The corresponding $\tilde{\mathbf{y}}(t)$ and $\bar{\mathbf{y}}(t)$ are defined by

$$\tilde{\mathbf{y}}(t) = \mathbf{G}\tilde{\mathbf{s}}(t) \qquad (9)$$

$$\bar{\mathbf{y}}(t) = \mathbf{G}\bar{\mathbf{s}}(t) \qquad (10)$$

Using the assumption that source signals $\mathbf{s}(t)$ are spatially independent and temporally IID, one can easily show that

$$\begin{aligned} E\{\Delta\mathbf{y}(t)\Delta\mathbf{y}^T(t)\} &= E\{\tilde{\mathbf{y}}(t)\tilde{\mathbf{y}}^T(t)\} + E\{\tilde{\mathbf{y}}(t-1)\tilde{\mathbf{y}}^T(t-1)\} \\ &= 2E\{[\mathbf{y}(t)-\bar{\mathbf{y}}(t)][\mathbf{y}(t)-\bar{\mathbf{y}}(t)]^T\} \qquad (11) \end{aligned}$$

Thus, when convergence of the algorithm in eqn. 4 is achieved, it satisfies

$$E\{[\mathbf{y}(t) - \bar{\mathbf{y}}(t)][\mathbf{y}(t) - \bar{\mathbf{y}}(t)]^T\} = \mathbf{0} \qquad (12)$$

In a similar manner, the modification of the algorithm in eqn. 3 for ICA is described by

$$\mathbf{W}(t+1) = \mathbf{W}(t) + \eta_t\{\mathbf{I} - \mathbf{f}(\Delta\mathbf{y}(t))\Delta\mathbf{y}^T(t)\}\mathbf{W}(t) \qquad (13)$$

It can easily be seen that for the nonlinear odd function $\mathbf{f}(\mathbf{y}(t))$, the stationary points of eqn. 13 satisfy

$$\begin{aligned} E\{\mathbf{f}(\Delta\mathbf{y}(t))\Delta\mathbf{y}^T(t)\} &= 2E\{\mathbf{f}(\tilde{\mathbf{y}}(t))\tilde{\mathbf{y}}^T(t)\} \\ &= \mathbf{0} \qquad (14) \end{aligned}$$

Computer simulation results: One exemplary simulation result is provided, for brevity. The observation vector $\mathbf{x}(t)$ was generated by

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) \qquad (15)$$

where the mixing matrix $\mathbf{A}$ is given by

$$\mathbf{A} = \begin{bmatrix} 0.129 & 0.504 \\ 0.605 & 0.951 \end{bmatrix} \qquad (16)$$

Two independent source signals $s_1(t)$ and $s_2(t)$ were drawn from a uniform distribution. Over the duration [1, 2000], the means of the source signals are $\bar{s}_1(t) = 0.267$ and $\bar{s}_2(t) = 0.452$. The mean of $s_1(t)$ and $s_2(t)$ were changed over the duration [2001, 10000] as follows: $\bar{s}_1(t) = -0.503$, $\bar{s}_2(t) = -0.417$.

As a performance measure, the following performance index PI was used. It is defined by

$$PI = \sum_{i=1}^{n}\left\{\left(\sum_{k=1}^{n}\frac{|g_{ik}|^2}{\max_j g_{ij}} - 1\right) + \left(\sum_{k=1}^{n}\frac{|g_{ki}|^2}{\max_j g_{ji}} - 1\right)\right\} \qquad (17)$$

where g_{ij} is the (i, j)th element of the global transformation matrix $\mathbf{G}$ and $\max_j g_{ij}$ represents the maximum value among the elements in the ith row vector of $\mathbf{G}$, $\max_j g_{ji}$ represents the maximum value among the elements in the ith column vector of $\mathbf{G}$. When perfect signal separation is carried out, the performance index PI is zero. In practice, it is a very small number.